

## WHAT IS CLAIMED IS:

1. A method for stack memory protection comprising the steps of:
- generating new memory page attributes for a page table used to manage memory, each of said new memory page attributes identifying a block of memory as a new class of memory, each of said new memory page attributes generated by a corresponding new load/store instruction;
  - assigning, by an operating system or a processor, a selected one of said new memory page attributes to a selected block of memory, said selected block of memory used as a new class of memory corresponding to said selected new memory page attribute;
  - blocking normal load /stores to a memory block having one of said new memory page attributes; and
  - blocking a first new load/store to a memory block with one of said new memory page attributes not corresponding to said first new load/store

7. The method of claim 5, wherein said first memory stack is a processor stack, said processor stack used by a processor to load and store hardware register contents

3 during program execution, said processor stacks transparent to a programmer or a  
4 compiler.

1 8. The method of claim 7, wherein said processor stack is an IA64 register stack.

1 9. The method of claim 5, wherein said second memory stack is a program stack,  
2 said program stack used by a programmer or a compiler in managing program flow.

004060 TET 45960

1        10.     A processor comprising stack memory protection circuitry, said processor  
2        using blocks of memory as stack memory, said stack memory protection circuitry  
3        comprising:

4                a stack memory attribute circuit, said stack memory attribute circuit  
5                operable to generate memory attributes, said memory attributes  
6                associated with each memory block designated as a memory stack;

7                a page table attribute storage circuit, said page table attribute circuit  
8                operable to store and associate one of said stack memory attributes  
9                with a block of memory designated as stack memory;

10               a stack memory allocation circuit, said stack memory allocation circuit  
11               operable to identify a block of memory as a stack memory and  
12               associate said memory block with one of said stack memory attributes,  
13               said stack memory attributes stored in a memory page table; and

14               a stack memory instruction execution circuit, said stack memory  
15               instruction execution circuit operable to decode load/store instructions  
16               to memory blocks, said stack memory instruction execution circuit  
17               granting stack memory load and stores to memory blocks having a

00657421-050700

096571090000

1        15.        The processor of claim 13, wherein said first memory stack is a processor  
2        stack, said processor stack used by a processor to load and store hardware register

4

1

2

3

0967131-090700

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

operable to identify a block of memory as a stack memory and  
associate said memory block with a stack memory attribute, said stack  
memory attribute stored in a memory page table; and

a stack memory instruction execution circuit, said stack memory  
instruction execution circuit operable to decode load/store instructions  
to memory blocks, said stack memory instruction execution circuit  
granting stack memory load and stores to memory blocks having a  
stack memory attribute and not granting stack memory load and stores  
to memory blocks not having said stack memory attribute.

18. The data processing system in claim 17, wherein a first error condition is  
generated whenever normal load/stores are attempted to stack memory having a first  
or a second stack memory attribute.

19. The data processing system in claim 17, wherein a second error condition is  
generated whenever said stack memory load/stores are attempted to memory not  
having a stack memory attribute.

20. The data processing system in claim 17, wherein a third error condition is  
generated whenever a stack memory load/store for a first memory stack is attempted

092100Z

**Figure 1**

[illegible]

	197	86
	198	86
	199	86
	200	86
	201	86
	202	86
	203	86
	204	86
	205	86
	206	86
	207	86
	208	86
	209	86
	210	86
	211	86
	212	86
	213	86
	214	86
	215	86
	216	86
	217	86
	218	86
	219	86
	220	86
	221	86
	222	86
	223	86
	224	86
	225	86
	226	86
	227	86
	228	86
	229	86
	230	86
	231	86
	232	86
	233	86
	234	86
	235	86
	236	86
	237	86
	238	86
	239	86
	240	86
	241	86
	242	86
	243	86
	244	86
	245	86
	246	86
	247	86
	248	86
	249	86
	250	86
	251	86
	252	86
	253	86
	254	86
	255	86
	256	86
	257	86
	258	86
	259	86
	260	86
	261	86
	262	86
	263	86
	264	86
	265	86
	266	86
	267	86
	268	86
	269	86
	270	86
	271	86
	272	86
	273	86
	274	86
	275	86
	276	86
	277	86
	278	86
	279	86
	280	86
	281	86
	282	86
	283	86
	284	86
	285	86
	286	86
	287	86
	288	86
	289	86
	290	86
	291	86
	292	86
	293	86
	294	86
	295	86
	296	86
	297	86
	298	86
	299	86
	300	86
	301	86
	302	86
	303	86
	304	86
	305	86
	306	86
	307	86
	308	86
	309	86
	310	86
	311	86
	312	86
	313	86
	314	86
	315	86
	316	86
	317	86
	318	86
	319	86
	320	86
	321	86
	322	86
	323	86
	324	86
	325	86
	326	86
	327	86
	328	86
	329	86
	330	86
	331	86
	332	86
	333	86
	334	86
	335	86
	336	86
	337	86
	338	86
	339	86
	340	86
	341	86
	342	86
	343	86
	344	86
	345	86
	346	86
	347	86
	348	86
	349	86
	350	86
	351	86
	352	86
	353	86
	354	86
	355	86
	356	86
	357	86
	358	86
	359	86
	360	86
	361	86
	362	86
	363	86
	364	86
	365	86
	366	86
	36	

1        24.    A computer program product embodied in a machine readable medium,  
2        including an operating system and a compiler for a processor system, comprising; a  
3        program of instructions for performing the program steps of:

4                generating new memory page attributes for a page table used to  
5                manage memory, each of said new memory page attributes identifying  
6                a block of memory as a new class of memory, each of said new  
7                memory page attributes generated by a corresponding new load/store  
8                instruction;

9                assigning, by an operating system or a processor, a selected one of said  
10                new memory page attributes to a selected block of memory, said  
11                selected block of memory used as a new class of memory  
12                corresponding to said selected new memory page attribute;

13                blocking normal load /stores to a memory block having one of said  
14                new memory page attributes; and

15                blocking a first new load/store to a memory block with one of said new  
16                memory page attributes not corresponding to said first new load/store

**0967-3102**

**Figure 1**

- 23 -

3 register contents during program execution, said processor stacks transparent to a  
4 programmer or a compiler.

1 31. The computer program product of claim 30, wherein said processor stack is an  
2 IA64 register stack.

1 32. The computer program product of claim 28, wherein said second memory  
2 stack is a program stack, said program stack used by a programmer or a compiler in  
3 managing program flow.

00/060" PAT 25960

1 33. A method of managing a memory device comprising the steps of:

2 partitioning said memory device into a plurality of memory spaces on  
3 an as-needed basis; and

4 associating a memory attribute with each memory space; said memory  
5 attribute determining a use of each of said memory spaces.

1 34. The method of claim 33, wherein a particular memory attribute has  
2 corresponding load/store instruction.

1 35. The method of claim 34, wherein a load/store instruction associated with a  
2 first memory attribute causes an error condition if attempted on a memory space with  
3 a second memory attribute.

1 36. The method of claim 33, wherein each of said memory attributes are stored in  
2 a memory page table, said memory page table used to manage said memory device.